

WHAT IS CLAIMED IS:

1. A method for designing a computer program,  
comprising:

5 accessing a plurality of domain rules, each domain  
rule being invariant;

displaying a plurality of business rules, each  
business rule being variable;

10 selecting one or more business rules of the  
plurality of business rules in response to a user  
selection;

customizing the one or more business rules;

associating the one or more business rules with a  
procedure;

associating the domain rules with the procedure;

15 displaying a model representing the procedure; and

generating a code corresponding to the procedure in  
order to design a computer program.

2. The method of Claim 1, further comprising:

20 collecting the domain rules and the business rules;

allocating the domain rules and the business rules  
to a plurality of use cases;

realizing the use cases; and

25 assessing the domain rules and the business rules in  
accordance with the realization.

3. The method of Claim 1, further comprising:

checking a syntax of the code; and

30 providing a notification if a syntax error is  
detected.

4. The method of Claim 1, further comprising:  
checking a logical consistency of the code; and  
providing a notification if a logical inconsistency  
is detected.

5

5. The method of Claim 1, further comprising:  
checking a compatibility between the model and the  
code; and  
providing a notification if an inconsistency is  
detected.

10

6. The method of Claim 1, wherein the model is  
expressed according to a common modeling language.

7. Logic for designing a computer program, the logic embodied in a medium and operable to:

access a plurality of domain rules, each domain rule being invariant;

5 display a plurality of business rules, each business rule being variable;

select one or more business rules of the plurality of business rules in response to a user selection;

customize the one or more business rules;

10 associate the one or more business rules with a procedure;

associate the domain rules with the procedure;

display a model representing the procedure; and

15 generate a code corresponding to the procedure in order to design a computer program.

8. The logic of Claim 7, further operable to:

collect the domain rules and the business rules;

20 allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

25 9. The logic of Claim 7, further operable to:

check a syntax of the code; and

provide a notification if a syntax error is detected.

10. The logic of Claim 7, further operable to:  
check a logical consistency of the code; and  
provide a notification if a logical inconsistency is  
detected.

5

11. The logic of Claim 7, further operable to:  
check a compatibility between the model and the  
code; and  
provide a notification if an inconsistency is  
detected.

10

12. The logic of Claim 7, wherein the model is  
expressed according to a common modeling language.

13. A system for designing a computer program,  
comprising:

a database operable to store a plurality of domain  
rules, each domain rule being invariant; and

5 a server coupled to the database and operable to:

display a plurality of business rules, each  
business rule being variable;

select one or more business rules of the  
plurality of business rules in response to a user  
10 selection;

customize the one or more business rules;

associate the one or more business rules with a  
procedure;

associate the domain rules with the procedure;

15 display a model representing the procedure; and

generate a code corresponding to the procedure  
in order to design a computer program.

14. The system of Claim 13, the server further  
20 operable to:

collect the domain rules and the business rules;

allocate the domain rules and the business rules to  
a plurality of use cases;

realize the use cases; and

25 assess the domain rules and the business rules in  
accordance with the realization.

15. The system of Claim 13, the server further  
operable to:

30 check a syntax of the code; and

provide a notification if a syntax error is  
detected.

16. The system of Claim 13, the server further operable to:

check a logical consistency of the code; and  
5 provide a notification if a logical inconsistency is detected.

17. The system of Claim 13, the server further operable to:

10 check a compatibility between the model and the code; and  
provide a notification if an inconsistency is detected.

15 18. The system of Claim 13, wherein the model is expressed according to a common modeling language.

19. A system for designing a computer program,  
comprising:

means for accessing a plurality of domain rules,  
each domain rule being invariant;

5 means for displaying a plurality of business rules,  
each business rule being variable;

means for selecting one or more business rules of  
the plurality of business rules in response to a user  
selection;

10 means for customizing the one or more business  
rules;

means for associating the one or more business rules  
with a procedure;

15 means for associating the domain rules with the  
procedure;

means for displaying a model representing the  
procedure; and

means for generating a code corresponding to the  
procedure in order to design a computer program.

20. A method for designing a computer program,  
comprising:

collecting a plurality of domain rules, allocating  
the domain rules to a plurality of use cases, realizing  
5 the use cases, assessing the domain rules in accordance  
with the realization, and accessing the domain rules,  
each domain rule being invariant;

displaying a plurality of business rules, each  
business rule being variable;

10 selecting one or more business rules of the  
plurality of business rules in response to a user  
selection;

customizing the one or more business rules;

15 associating the one or more business rules with a  
procedure;

associating the domain rules with the procedure;

displaying a model representing the procedure, the  
model expressed according to a common modeling language;

20 generating a code corresponding to the procedure in  
order to design a computer program;

checking a syntax of the code, and providing a  
notification if a syntax error is detected;

25 checking a logical consistency of the code, and  
providing a notification if a logical inconsistency is  
detected; and

checking a compatibility between the model and the  
code, and providing a notification if an inconsistency is  
detected.



21. A method for managing rules for designing a computer program, comprising:

accessing a plurality of rules;

5 analyzing the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

storing the business rules; and

10 providing a business rule from the stored business rules in response to a request for the business rule.

22. The method of Claim 21, further comprising:

customizing the provided business rule;

15 associating the customized business rule with a procedure; and

generating a code corresponding to the procedure in order to design a computer program.

23. The method of Claim 21, further comprising:

20 associating the domain rules with a procedure; and

generating a code corresponding to the procedure in order to design a computer program.

24. The method of Claim 21, further comprising:

25 allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases; and

assessing the domain rules and the business rules in accordance with the realization.

25. A system for managing rules for designing a computer program, comprising:

a database operable to store a plurality of rules;  
and

5 a server coupled to the database and operable to:

analyze the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

10 store the business rules; and

provide a business rule from the stored business rules in response to a request for the business rule.

15 26. The system of Claim 25, wherein the server is further operable to:

customize the provided business rule;

associate the customized business rule with a procedure; and

20 generate a code corresponding to the procedure in order to design a computer program.

27. The system of Claim 25, wherein the server is further operable to:

25 associate the domain rules with a procedure; and

generate a code corresponding to the procedure in order to design a computer program.

28. The system of Claim 25, wherein the server is further operable to:

allocate the domain rules and the business rules to a plurality of use cases;

5 realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

29. Logic for managing rules for designing a computer program, the logic embodied in a medium and operable to:

access a plurality of rules;

5 analyze the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

store the business rules; and

10 provide a business rule from the stored business rules in response to a request for the business rule.

30. The logic of Claim 29, further operable to:

customize the provided business rule;

15 associate the customized business rule with a procedure; and

generate a code corresponding to the procedure in order to design a computer program.

31. The logic of Claim 29, further operable to:

20 associate the domain rules with a procedure; and

generate a code corresponding to the procedure in order to design a computer program.

32. The logic of Claim 29, further operable to:

25 allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

33. A system for managing rules for designing a computer program, comprising:

means for accessing a plurality of rules;

5 means for analyzing the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

means for storing the business rules; and

10 means for providing a business rule from the stored business rules in response to a request for the business rule.

34. A method for managing rules for designing a computer program, comprising:

accessing a plurality of rules;

5 analyzing the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

allocating the domain rules and the business rules to a plurality of use cases;

10 realizing the use cases;

assessing the domain rules and the business rules in accordance with the realization;

storing the business rules;

15 providing a business rule from the stored business rules in response to a request for the business rule;

customizing the provided business rule;

associating the customized business rule with a procedure;

associating the domain rules with the procedure; and

20 generating a code corresponding to the procedure in order to design a computer program.